Creating logical volumes (LVM - Logical Volume Management) is a way to manage disk storage that is flexible and more efficient than traditional partitioning. The process to create logical volumes is fairly similar across Linux distributions like **Ubuntu** and **CentOS**. Below is a step-by-step guide for creating logical volumes with commands that are compatible with both **Ubuntu** and **CentOS**.

### **Prerequisites:**

* You need to have **root** or **sudo** privileges.
* Install LVM tools if they are not already installed:

For **Ubuntu**:  
 sudo apt update

sudo apt install lvm2

For **CentOS**:  
 sudo yum install lvm2

### **Step 1: Prepare the Physical Volume (PV)**

A physical volume is typically a hard disk, partition, or a disk array.

1. **Identify the available disks**:

Run the following command to list the available disks:  
  
 lsblk

Alternatively, you can use:  
  
 fdisk -l

1. **Create a physical volume**:

Convert a physical device (e.g., /dev/sdb) into a physical volume.  
  
 sudo pvcreate /dev/sdb

Verify the physical volume creation:  
  
 sudo pvdisplay

### **Step 2: Create a Volume Group (VG)**

A Volume Group is a pool of storage that consists of one or more physical volumes.

1. **Create the volume group**:

Create a volume group by naming it (e.g., my\_volume\_group) and associating it with the physical volume (/dev/sdb).  
 sudo vgcreate my\_volume\_group /dev/sdb

1. **Verify the volume group**:

List all volume groups and their sizes:  
 sudo vgdisplay

### **Step 3: Create a Logical Volume (LV)**

A logical volume is a virtual partition that resides within a volume group.

1. **Create a logical volume**:

To create a logical volume named my\_logical\_volume with a size of 10 GB from the volume group my\_volume\_group, use:  
  
 sudo lvcreate -L 10G -n my\_logical\_volume my\_volume\_group

* + Here:  
    - -L 10G specifies the size of the logical volume.
    - -n my\_logical\_volume specifies the name of the logical volume.
    - my\_volume\_group is the name of the volume group.

1. **Verify the logical volume**:

To display details of the logical volume:  
 sudo lvdisplay

### **Step 4: Format the Logical Volume with a Filesystem**

After creating a logical volume, it needs to be formatted with a filesystem before it can be used.

1. **Format the logical volume**:

For example, using **ext4** filesystem:  
  
 sudo mkfs.ext4 /dev/my\_volume\_group/my\_logical\_volume

Alternatively, you can use other filesystems like **xfs**:  
  
 sudo mkfs.xfs /dev/my\_volume\_group/my\_logical\_volume

1. **Verify the filesystem**:

To check the filesystem:  
 sudo blkid /dev/my\_volume\_group/my\_logical\_volume

### **Step 5: Mount the Logical Volume**

1. **Create a mount point**:

Create a directory where the logical volume will be mounted:  
 sudo mkdir /mnt/my\_logical\_volume

1. **Mount the logical volume**:

Mount the logical volume to the created directory:  
 sudo mount /dev/my\_volume\_group/my\_logical\_volume /mnt/my\_logical\_volume

1. **Verify the mount**:

Use df -h to verify the mount:  
 df -h

### **Step 6: Configure Auto-mounting (Optional)**

To ensure that the logical volume mounts automatically on boot, add an entry in the /etc/fstab file.

1. **Get the UUID of the logical volume**:

Run the following command to get the UUID:  
  
 sudo blkid /dev/my\_volume\_group/my\_logical\_volume

Example output:  
  
 /dev/my\_volume\_group/my\_logical\_volume: UUID="xxxx-xxxx" TYPE="ext4"

1. **Edit /etc/fstab**:

Open the /etc/fstab file in a text editor:  
 sudo nano /etc/fstab

1. **Add an entry**:

Add the following line at the end of the file:  
 UUID=xxxx-xxxx /mnt/my\_logical\_volume ext4 defaults 0 2

1. **Save and close**:  
   * Save the file and close the editor.
2. **Test fstab**:

You can test if the fstab is configured correctly by running:  
 sudo mount -a

### **Step 7: Extend Logical Volume (Optional)**

If you need more space in your logical volume, you can extend it.

1. **Extend the logical volume**:

For example, to add 5 GB to my\_logical\_volume:  
 sudo lvextend -L +5G /dev/my\_volume\_group/my\_logical\_volume

1. **Resize the filesystem**:  
   * After extending the logical volume, resize the filesystem to use the new space:

For **ext4** filesystem:  
  
 sudo resize2fs /dev/my\_volume\_group/my\_logical\_volume

For **xfs** filesystem:  
  
 sudo xfs\_growfs /dev/my\_volume\_group/my\_logical\_volume

### **Step 8: Remove a Logical Volume (Optional)**

If you no longer need the logical volume, you can remove it.

**Unmount the logical volume**:  
  
 sudo umount /mnt/my\_logical\_volume

**Remove the logical volume**:  
  
 sudo lvremove /dev/my\_volume\_group/my\_logical\_volume

**Remove the volume group (optional)**: If you no longer need the volume group, you can remove it:  
  
 sudo vgremove my\_volume\_group

**Remove the physical volume (optional)**: If you no longer need the physical volume, you can remove it:  
  
 sudo pvremove /dev/sdb

### **Conclusion:**

This guide walks you through creating, formatting, mounting, and managing logical volumes using **LVM** on both **Ubuntu** and **CentOS**. The commands are the same across both distributions, and this guide ensures that your logical volume management is efficient and works across multiple systems.